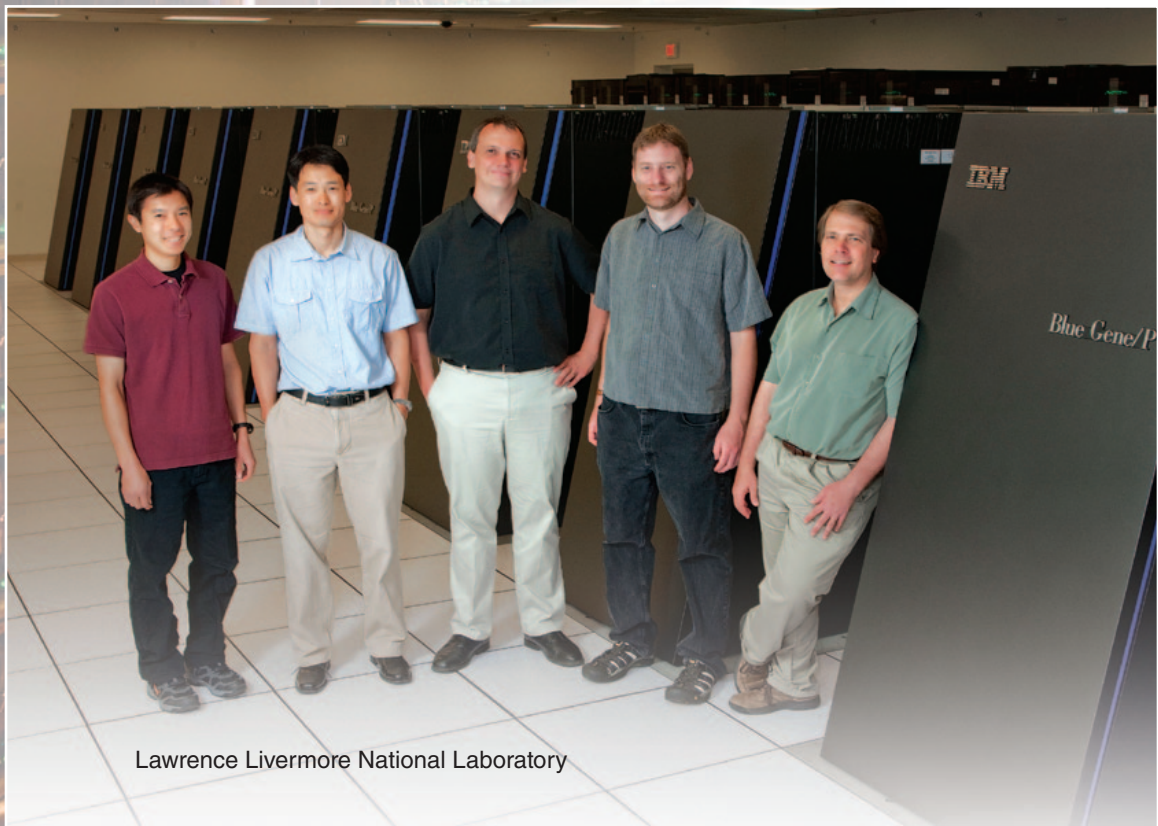


# Lightweight, Scalable Tool Identifies Supercomputers' Code Errors

**C**ONSIDER the common desktop computer. The hardware that runs all of an application's many independent execution units, or processes, is called a core. Most desktops have between one and four cores. These computers cost between a few hundred and a few thousand dollars. In contrast, today's largest supercomputers contain hundreds of thousands of cores and cost hundreds of millions of dollars. The scientific applications these computers run—which often culminate from decades of multiperson development efforts—are equally costly. Single faults, or bugs in the codes, that disable only one process can halt an application's execution. During the resulting debugging process, developers use up their time locating the bugs, with delays consuming machine hours and racking up significant costs.

Extreme-scale systems present an additional hurdle. Current debugging tools were never designed to scale to such sizes. Therefore, at scales of a thousand processes, these tools can take minutes to perform a single debugging operation, and typically, each operation is performed tens to hundreds of times during just one debug session. "Our debugging tool is our response to this problem," says computer scientist Greg Lee. Lee and fellow Livermore computer scientists Dong Ahn, Bronis de Supinski, Matthew LeGendre, and Martin Schulz, with collaborators at the University of Wisconsin at Madison and the University of New Mexico, designed and developed a unique R&D Award-winning solution called the stack trace analysis tool (STAT). The tool can identify errors in code running on today's largest machines. It will also work on the even larger machines expected to roll out over the next several years.



Lawrence Livermore National Laboratory

“The approach of other debuggers provides so much detailed information about each process that they are inherently unusable at such extreme scales,” says Lee. STAT, on the other hand, was designed to provide meaningful information quickly. “Just as medical staff can call ‘STAT’ to get immediate action and help patients in distress, when users of supercomputers need to debug an application at extreme scales, they can call on STAT,” says Lee.

## Getting Help STAT

Errors in computer codes arise not only in the initial development phase of applications, when the code is beginning to take shape, but also when new features are added to mature software. These bugs sometimes lay dormant even in heavily tested and widely used codes, only to emerge when run with a new data set, on a new platform, or at larger scales. Scientific codes designed for high-performance computing systems provide additional challenges because their complex codes incorporate multiple mathematical and scientific software libraries.

STAT works by detecting and grouping similar processes at suspicious points in an application's execution. It quickly and automatically identifies anomalies and outliers—processes that cannot be grouped or whose behavior is substantially different—because they often indicate flawed execution. STAT achieves this grouping by dynamically examining the state of each process and extracting the call stacks—the sequence of function calls—that led to the current point of execution. In this way, STAT can relate the state of the processes to each other.

## Speed Daemons

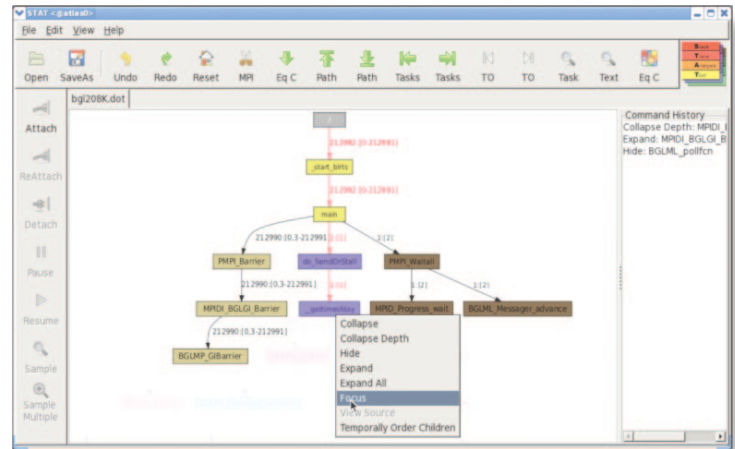
STAT offers varying levels of detail in the call stacks, from coarser function granularity to specific source-code line numbers. Because it gathers stack traces across the entire application, it provides a global view of what every process is doing. These stack traces are merged to reduce the problem search space, so users can identify a small yet representative subset of tasks on which to apply heavyweight analysis.

Another important scaling advantage that STAT has over similar tools is its lightweight design, which allows the tool to maintain interactive response times. Most scientific applications make full use of all available processing power and memory capacity, which leaves few resources for tools. STAT's daemons—tool processes that run alongside the application—have very low computational and memory requirements.

## Pinpointing Problems

STAT can not only distinguish a process that is stuck in a single location in the code but also pinpoint the exact task

The Livermore development team for STAT: (from left) Greg Lee, Dong Ahn, Martin Schulz, Matthew LeGendre, and Bronis de Supinski.



The stack trace analysis tool (STAT) includes a powerful and intuitive graphical user interface that allows the user to identify quickly where a bug exists in an application. STAT automatically analyzes the state of the application and pinpoints potential bug locations.

causing the hang. STAT also derives the relative execution progress of each application task, which is useful for determining problematic application processes. “A culprit may have made the least execution progress through the code because it’s stuck in a computation phase that the rest of the application processes have already passed,” Lee says.

The tool has run on a wide range of supercomputer platforms, including the IBM BlueGene family of machines and several of the world's fastest as reported by the Top500 Supercomputer Sites list. STAT also runs on the Cray XT and Cray XE high-performance computers and has been demonstrated at 216,000 cores on Oak Ridge National Laboratory's Jaguar system, which once reigned as the fastest supercomputer. STAT was recently run to discover a bad node on IBM BlueGene/L. "In this case, STAT definitely proved itself useful," says Lee. "At best, finding this problem would have bordered on impossible without STAT."

The team is excited about the recognition and publicity the award will bring and looks forward to helping others adopt the tool. In the future, the scientists hope to complete their research and turn STAT over to a company for commercialization.

—Kris Fury

**Key Words:** bug, computer core, debugging, extreme-scale computing system, high-performance computing, R&D 100 Award, stack trace analysis tool (STAT).

**For further information contact Greg Lee (925) 424-6554 (lee218@ltnl.gov).**